Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

1. (Previously Presented) A method of transport protocol optimization of an internet protocol, comprising the steps of (a) using a source packet interceptor to intercept an IP packet from a source application, the source packet interceptor examines an IP header of the IP packet to determine if it is an IP packet to be intercepted, (b) using a source edge process to act as the new destination for the source application, (c) using a source packet driver to aggregate the intercepted IP packets from the source application, (d) using a source data mover to transport the aggregated IP packets over a communication link to a destination data mover, (e) using a destination packet driver to disaggregate the transported aggregated packets, (f) using a destination edge process to deliver the disaggregated IP packets to a destination application.

2. (Original) The transport protocol optimization method of claim 1, comprising the step of using IP routing.

3. (Previously Presented) The transport protocol optimization method of claim 1, the IP packet comprising a TCP, UDP, ICMP, or other type of IP packet.

4. (Previously Presented) The transport protocol optimization method of claim 1, the step of intercepting an IP packet from the source application comprises the steps of comparing the IP packet's address to packet addresses in a look-up table and (b) intercepting only those source packets with the same addresses as those stored in the look-up table.

5. (Previously Presented) The transport protocol optimization method of claim 1, the address of the IP packet comprises the packet's source IP address, source port number, destination IP address, destination port number, and protocol type.

6. (Previously Presented) The transport protocol optimization method of claim 1, the step of intercepting an IP packet from the source application comprises the step of routing the IP packet to an edge process that is unique to the address of the IP packet.

7. (Previously Presented) The transport protocol optimization method of claim 1, the step of intercepting an IP packet from the source application comprises the steps of a source edge process (a) reading the data contained in the routed IP packets and (b) forming a message header field for the routed IP packets.

8. (Original) The transport protocol optimization method of claim 1, comprising the step of the packet driver forming a packet driver message.

9. (Previously Presented) The transport protocol optimization method of claim 8, the packet driver message comprises the message header field and intercepted IP packet data from the source edge process.

10. (Original) The transport protocol optimization method of claim 9, comprising the step of forming a plurality of packet driver messages.

11. (Original) The transport protocol optimization method of claim 10, comprising the step of aggregating multiple packet driver messages into a packet driver buffer.

12. (Previously Presented) The transport protocol optimization method of claim 11, the size of the aggregated packet driver messages is less than or equal to a predetermined maximum size of the buffer.

13. (Original) The transport protocol optimization method of claim 12, comprising the step of the packet driver forming a routing header in the packet driver buffer that precedes a first packet driver message.

14. (Previously Presented) The transport protocol optimization method of claim 13, the routing header comprises a function type field, a number of packet driver messages field, and a data length field.

15. (Previously Presented) The transport protocol optimization method of claim 7, the message header comprises a version field, a length of header field, a message function type field, a message flag field, a protocol type field, a sequence number field, a source IP address field, a destination IP address field, a source IP port number field, a destination IP port number field, a length of data field, and a status field.

16. (Original) The transport protocol optimization method of claim 6, comprising the step of combining a routing header field, a message header field, and the intercepted IP packet data from the edge process.

17. (Original) The transport protocol optimization method of claim 11, comprising the step of using a compression engine to compress the aggregated packet driver messages.

18. (Original) The transport protocol optimization method of claim 17, comprising the step of routing the aggregated packet driver messages to the source data mover.

19. (Previously Presented) The transport protocol optimization method of claim 18, transmission of packet driver buffers over a communication link by the data mover comprises one or more of the steps of (a) inserting data mover fields into the start of the packet driver buffer; (b) reducing the size of the packet driver buffer by breaking the buffer into multiple segments, with each segment being no greater than the size specified in the configuration file; (c) using standard UDP socket calls to interface with the TCP stack for UDP delivery of the segments over the network.

20. (Previously Presented) The transport protocol optimization method of claim 19, the communication link comprising a TCP, UDP, or other TCP/IP link.

21. (Previously Presented) The transport protocol optimization method of claim 19, the data mover protocol comprising (a) data mover transport data subfield, and (b) data mover transport acknowledgement subfield.

22. (Previously Presented) The transport protocol optimization method of claim 21, the data mover transport data subfield comprising the length of the entire subfield, the subfield type code, the logical sequence number of the packet driver buffer or packet drive buffer segment, and the physical sequence number of the packet driver buffer or packet driver buffer segment.

23. (Previously Presented) The transport protocol optimization method of claim 21, the data mover transport acknowledgement subfield comprising the length of the entire subfield, the subfield type code, the highest physical block number sent from the source packet driver to the destination packet driver over a communication link, the highest physical block number received by the source packet driver that was sent by the destination packet driver over a communication link, the bit-significant flags representing the blocks received, and the rate of data delivery to the destination packet driver.

24. (Previously Presented) The transport protocol optimization method of claim 1, the packets intercepted by an operating system exit point.

25. (Original) The transport protocol optimization method of claim 4 comprising the step of modifying the destination address of the IP packets accepted for interception to be the address of the source packet interceptor.

26. (Original) The transport protocol optimization method of claim 6, comprising the step of creating a edge process for each TCP application connection; a UDP edge process for each UDP intercept; and a ICMP edge process for a ICMP intercept.

27. (Original) The transport protocol optimization method of claim 1, comprising the step of terminating any connection between a source application and a destination application.

28. (Original) The transport protocol optimization method of claim 1, comprising the step of opening a connection between a source application and a destination data application.

29. (Original) The transport protocol optimization method of claim 28, comprising the steps of (a) opening a connection between the source application and the source edge processor and (b) opening a connection between the destination edge processor and the destination application.

30. (Previously Presented) The transport protocol optimization method of claim 20, the TCP, UDP, or other TCP/IP link for transporting the stored packets is over a WAN.

31. (Original) The transport protocol optimization method of claim 29, comprising the steps of (a) transporting packets from the source application to the source packet interceptor over a source LAN and (b) transporting packets delivered to a destination data mover to a destination application over a destination LAN.

32. (Previously Presented) The transport protocol optimization method of claim 17, a decompression engine performs the step of decompressing the aggregated packet driver messages.

33. (Previously Presented) The transport protocol optimization method of claim 1, optimization is comprised of the step of using transport protocol optimization source software and destination software.

34. (Previously Presented) The transport protocol optimization method of claim 33, the source software runs on a source server, a source network switch, or as a source network appliance and the destination software runs on a destination server, a destination network switch, or as a destination network appliance.

35. (Previously Presented) The transport protocol optimization method of claim 34, comprising the step of connecting the source and destination network appliances to a (a) network switch, which switch is connected to an application server running a application; (b) network switch, which switch is connected to an application server running a application and to a network router; or (c) to an application server running a application.

36. (Original) The transport protocol optimization method of claim 1, comprising the step of integrating the source packet interceptor, packet driver, edge process, and data mover into a source TPO.

37. (Original) The transport protocol optimization method of claim 1, comprising the step of integrating the packet interceptor, packet driver, edge process, and data mover into a destination TPO.

38. (Original) The transport protocol optimization method of claim 1, comprising the step of using a source TPO and a destination TPO to create a pair of TPOs.

39. (Previously Presented) The transport protocol optimization method of claim 38, comprising a plurality of pairs of TPOs for multicasting and for multipoint communication.

40. (Original) The transport protocol optimization method of claim 1, comprising the steps of (a) attaching a source server running the source application on a source LAN, (b) attaching a source TPO on the source LAN and, (c) attaching a destination server running a destination application on a destination LAN, and (d) attaching a destination TPO on the destination LAN.

41. (Previously Presented) The transport protocol optimization method of claim 40, the packets from the source application are transported over the source LAN to the source TPO and the packets from the destination TPO are transported over the destination LAN to the destination application.

42. (Original) A method of internet protocol optimization, comprising the steps of: (a) using a packet interceptor to intercept an IP packet identified in a look-up table as having a specified source address, source port number, destination address, destination port number, and protocol type; (b) using a packet driver to encapsulate the IP packet into a packet driver message, to aggregate packet driver messages, and to route aggregated packet drive messages to a data mover; (c) using the data mover to route via IP routing the aggregated packet driver messages to a second data mover over a UDP communication link; (d) using the second data mover to route the aggregated packet driver messages to a second packet driver; (e) using a second packet driver

to disaggregate the packet driver messages and to dis-encapsulate the IP packet; and (f) using a second edge process to deliver the dis-encapsulated IP packet to a destination.

43-45. (Canceled)

46. (Previously Presented) A device for internet protocol optimization, comprising means for: (a) examination of IP packets to identify packets to be intercepted and interception of such packets, ; (b) encapsulating the intercepted packet with a message header field; (c) encapsulating the message header field with a packet driver message; (d) aggregating packet driver messages; (e) delivering aggregated packet driver messages to a data mover; (f) transporting aggregated packet driver messages to a destination; (g) disaggregating transported packets; (h) de-encapsulating the message header field; (j) deencapsulating the intercepted packet; and (i) delivering the packets to their respective destinations.